



# UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/624,808	07/22/2003	David Alan Bailey	ROC920030220US1	6643
46296 7590 07/16/2007 MARTIN & ASSOCIATES, LLC P.O. BOX 548 CARTHAGE, MO 64836-0548			EXAMINER DOAN, DUC T	
			ART UNIT 2188	PAPER NUMBER
			MAIL DATE 07/16/2007	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

Application No.

10/624,808

Applicant(s)

BAILEY ET AL.

Examiner

Duc T. Doan

Art Unit

2188

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 27 June 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 4,5,9,10,16,19 and 22-24 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 4-5,9-10,16,19,22-24 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## **DETAILED ACTION**

### ***Status of Claims***

Claims 1-19 have been presented for examination in this application. In response to the last office action, claims 22-24 have been added. Claims 1-3,6-8,11-15,17-18,20-21 have been cancelled. As the result, claims 4-5,9-10,16,19,22-24 are pending in this application.

Claims 4-5,9-10,16,19,22-24 are rejected.

Applicant's remarks/amendments filed 6/27/07 have been fully considered but they are not persuasive. Therefore, the rejections from the previous office action are respectfully maintained with changes as needed to address the amendments.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 4-5,9-10,16,19,22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Day et al (US Pub 2003/0084030) in view of Tarui et al (US Pub 2002/0112102), and further in view of Kaneko (2003/0163641).

As in claim 4, Day discloses an apparatus comprising: at least one processor (Day's Fig 1: #101A-H CPU); a memory coupled to the at least one processor (Day's Fig 1: #102 memory); a plurality of logical partitions defined on the apparatus (Day's Fig 2: #206B, #205B, #204B a hypervisor partition represented by user application code modules, operating system code modules such as high level OS and OS kernel),

a partition manager residing in the memory and executed by the at least one processor (Day's Fig 2: #203, #202 hypervisor management code and based hypervisor code PLIC executed by processor in memory, corresponding to the claim's partition manager),

the partition manager managing the plurality of logical partitions and executing separately from the plurality of logical partitions (Day's paragraph 28,37 discloses the partition management code (Fig 2: #203, #202 hypervisor management code and based hyper visor code PLIC) executes separately from logical partition to manage resources in plurality of logical partitions),

Day does not expressly disclose the claim's detail of I/O reconfiguration mechanism. However, Tarui discloses the partition manager (Tarui's paragraph 59 lines 1-2, partition control program such as hypervisor) comprising

an I/O reconfiguration mechanism that reconfigures identified I/O (Tarui's paragraph 53 reconfiguring I/O resources that being shared among logical partitions); and a logical partition suspend/resume mechanism that suspends at least one of the plurality of logical partitions before the I/O reconfiguration mechanism reconfigures the identified I/O by inhibiting dispatch of tasks to the at least one logic partition (Tarui's paragraph 84 discloses the partition control program instructs the OS of the current partitions (i.e the partitions whose allocations of resources are being changed, paragraph 83) to stop using the resources, thus obviously no more i/o tasks are

issued in these current partitions (i.e corresponding to the claim's suspending logical partition and inhibit dispatch of tasks mechanism);

and waiting until all pending tasks in the at least one logical partition are complete (Tarui's Fig 1: #101 discloses pending tasks in logical partition are flushed by the I/O adapter circuitry until it completed, when no more pending I/O request and byte counter value is zero, see Tarui's paragraphs 65, 67), and that resumes all suspended logical partitions after the I/O reconfiguration mechanism reconfigures the identified I/O by enable dispatching of tasks to the at least one logical partition (Tarui's paragraph 83 and 84 clearly teaches that after reconfiguration of i/o resources, the partitions are permitted/resumed to use the i/o resources, see Tarui's paragraph 84 lines 11-16), obviously when the partitions are permitted to use the i/o resource, the i/o tasks will be dispatched to the previously stopped logical partitions.

It would have been obvious to one of ordinary skill in the art at the time of invention to include the partition control program and associating logic as suggested by Tarui in Day's system thereby the reconfiguration of i/o resources can be done dynamically and in an automatic manner (Tarui's paragraph 84 lines 12-16);

Day does not expressly disclose the claim's aspect of ownership of I/O. However, Tarui further discloses the plurality of logical partitions comprising at least one logical partition that owns identified I/O and at least one logical partition that does not own the identified I/O (Tarui's paragraph 53, space sharing allocation allocates each partition with separate I/O resources; It's further noted that each logical partition can own and process its own identified I/O tasks in separate manner, thus a second logical partition would not process the I/O task that has been assigned, owned and identified by a first logical partition);

Day and Tarui do not expressly disclose the claim's aspects of suspends/resumes **all** of the plurality of the logical partitions. However, Kaneko's paragraph 31 discloses a storage subsystem with I/O resources being assigned separately to plurality of the logical partitions. Kaneko further discloses an I/O reconfiguration including **all** of the logical partitions are suspended, the I/O reconfiguration reconfigures the identified I/O by inhibiting dispatch of tasks to **all** of the plurality of logical partition (Kaneko's paragraph 32, the whole system is switch to a disconnect standby mode) and waiting until **all** pending tasks in all of the plurality of logical partitions are completed (Kaneko's paragraph 32 lines 13-14, when the requests from the host have been completely processed), and that resumes all of the plurality of logical partitions after the I/O reconfiguration mechanism reconfigures the identified I/O by enable dispatching of tasks to **all** of the plurality of logical partition (Kaneko's paragraph 33 discloses when there is no more left requests from the host, the logical partitioning configuring/changing is carried out, and when this step is completed, the system resumes to executing **all** suspended requests and other new dispatching requests). Kaneko's paragraph 24 further teaches that the logical partitioning of resources can be done in a dynamically manner. It would have been obvious to one of ordinary skill in the art at the time of invention to include the logical partitioning method and associating logic as suggested by Kaneko in Day's system modified by Tarui, thereby the reconfiguration of i/o resources can be done easily in a dynamically manner (Kaneko's paragraph 24).

Tarui's paragraph 83 further discloses the circuitry detects identified I/O required reconfiguration (reconfiguration changes for identified I/O tasks required when hot-plugged i/o resources occurring).

Claims 5,9-10,16,19 are rejected based on the same rationale as in claim 4.

As in claim 24, Day discloses an apparatus comprising: at least one processor (Day's Fig 1: #101A-H CPU); a memory coupled to the at least one processor (Day's Fig 1: #102 memory); a plurality of logical partitions defined on the apparatus (Day's Fig 2: #206B, #205B, #204B a hypervisor partition represented by user application code modules, operating system code modules such as high level OS and OS kernel),

a partition manager residing in the memory and executed by the at least one processor (Day's Fig 2: #203, #202 hypervisor management code and based hypervisor code PLIC executed by processor in memory, corresponding to the claim's partition manager),

the partition manager managing the plurality of logical partitions and executing separately from the plurality of logical partitions (Day's paragraph 28,37 discloses the partition management code (Fig 2: #203, #202 hypervisor management code and based hyper visor code PLIC) executes separately from logical partition to manage resources in plurality of logical partitions),

Day does not expressly disclose the claim's steps associating with the partition manager.

However, Tarui discloses the partition manager (Tarui's paragraph 59 lines 1-2, partition control program such as hypervisor) comprising:

an I/O reconfiguration mechanism that reconfigures identified I/O (Tarui's paragraph 53 reconfiguring I/O resources that being shared among logical partitions/identified tasks); and a logical partition suspend/resume mechanism that suspends at least one of the plurality of logical

partitions before the I/O reconfiguration mechanism reconfigures the identified I/O by inhibiting dispatch of tasks to the at least one logic partition (Tarui's paragraph 84 discloses the partition control program instructs the OS of the current partitions (i.e the partitions whose allocations of resources are being changed, paragraph 83) to stop using the resources, thus obviously no more i/o tasks are issued in these current partitions (i.e corresponding to the claim's suspending logical partition and inhibit dispatch of tasks mechanism to i/o resources, thus quiescing i/o resources associating with the tasks being suspended);

and waiting until all pending tasks in the at least one logical partition are complete (Tarui's Fig 1: #101 discloses pending tasks in logical partition are flushed by the I/O adapter circuitry until it completed, when no more pending I/O request and byte counter value is zero, see Tarui's paragraphs 65, 67), and that resumes all suspended logical partitions after the I/O reconfiguration mechanism reconfigures the identified I/O by enable dispatching of tasks to the at least one logical partition (Tarui's paragraph 83 and 84 clearly teaches that after reconfiguration of i/o resources, the partitions are permitted/resumed to use the i/o resources, see Tarui's paragraph 84 lines 11-16), obviously when the partitions are permitted to use the i/o resource, the i/o tasks will be dispatched to the previously stopped logical partitions.

It would have been obvious to one of ordinary skill in the art at the time of invention to include the partition control program and associating logic as suggested by Tarui in Day's system thereby the reconfiguration of i/o resources can be done dynamically and in an automatic manner (Tarui's paragraph 84 lines 12-16);

Day does not expressly disclose the claim's aspect of ownership of I/O. However, Tarui further discloses the plurality of logical partitions comprising at least one logical partition that



owns identified I/O and at least one logical partition that does not own the identified I/O (Tarui's paragraph 53, space sharing allocation allocates each partition with separate I/O resources; It's further noted that each logical partition can own and process its own identified I/O tasks in separate manner, thus a second logical partition would not process the I/O task that has been assigned, owned and identified by a first logical partition);

Day and Tarui do not expressly disclose the claim's details associating with I/O resources being arranged in I/O loops. However, Kanano discloses a storage subsystem with I/O resources that are arranged including a plurality of I/O towers (Kanano's Fig 2: #11 to #14, paragraphs 27-28) coupled to the apparatus via plurality of I/O loops (Kanano's Fig 2: #211 to #214, and #221 to #224 etc..arrays of disks are arranged in several I/O loops associating with towers Fig 2: #11 to #14, see Kanano's paragraphs 27-28 ). Kaneko further discloses an I/O reconfiguration mechanism including : detecting when the first I/O loop is unbalanced (Kaneko's paragraph 38, when a new disk resource being added into the storage subsystem; quiescing I/O resources in the first loop, determining which of the plurality of logical partitions own the I/O resources in the first loop, suspending the logical partitions determined (Kaneko's Fig 4, paragraphs 27-28,35, the requests which are received from the host by any other disk array unit (for example disk array unit 4) that accessing the resources of disk array unit #3 are determined, disconnected and suspended, for example to add a disk in the disk array unit 3); rebalancing the first I/O loop by allocating at least one I/O resource in the first loop from the first logical partition to the second logical partition (Kaneko's paragraph 38 further discloses an embodiment of rebalancing the first I/O loop, by producing a plurality of copies of disk data and dividing them so as to utilize the disk regions connected to the newly installed disk array unit); Obviously

after the rebalancing step, the new disk being added is ready for enable the I/O in the first loop after rebalancing; and resuming the logical partitions suspended .

It would have been obvious to one of ordinary skill in the art at the time of invention to include the logical partitioning method and associating logic as suggested by Kaneko in Day's system modified by Tarui, thereby the reconfiguration of i/o resources can be done easily thus further avoid the not intent errors/mistakes when partitioning are reconfiguring I/O resources for a network storage system (see Kaneko's paragraphs 8,6).

Claim 23 is rejected by the same rationale as of claim 22.

As in claim 24, Day discloses a computer readable program product comprising (Day's paragraph 70):

the partition manager managing the plurality of logical partitions and executing separately from the plurality of logical partitions (Day's paragraph 28,37 discloses the partition management code (Fig 2: #203, #202 hypervisor management code and based hyper visor code PLIC) executes separately from logical partition to manage resources in plurality of logical partitions), Day does not expressly disclose the claim's steps associating with the tasks being performed by the partition manager. However, Tarui discloses the partition manager (Tarui's paragraph 59 lines 1-2, partition control program such as hypervisor) comprising:

an I/O reconfiguration mechanism that reconfigures identified I/O (Tarui's paragraph 53 reconfiguring I/O resources that being shared among logical partitions/identified tasks); and a logical partition suspend/resume mechanism that suspends at least one of the plurality of logical partitions before the I/O reconfiguration mechanism reconfigures the identified I/O by inhibiting

dispatch of tasks to the at least one logic partition (Tarui's paragraph 84 discloses the partition control program instructs the OS of the current partitions (i.e the partitions whose allocations of resources are being changed, paragraph 83) to stop using the resources, thus obviously no more i/o tasks are issued in these current partitions (i.e corresponding to the claim's suspending logical partition and inhibit dispatch of tasks mechanism to i/o resources, thus quiescing i/o resources associating with the tasks being suspended);

and waiting until all pending tasks in the at least one logical partition are complete (Tarui's Fig 1: #101 discloses pending tasks in logical partition are flushed by the I/O adapter circuitry until it completed, when no more pending I/O request and byte counter value is zero, see Tarui's paragraphs 65, 67), and that resumes all suspended logical partitions after the I/O reconfiguration mechanism reconfigures the identified I/O by enable dispatching of tasks to the at least one logical partition (Tarui's paragraph 83 and 84 clearly teaches that after reconfiguration of i/o resources, the partitions are permitted/resumed to use the i/o resources, see Tarui's paragraph 84 lines 11-16), obviously when the partitions are permitted to use the i/o resource, the i/o tasks will be dispatched to the previously stopped logical partitions.

It would have been obvious to one of ordinary skill in the art at the time of invention to include the partition control program and associating logic as suggested by Tarui in Day's system thereby the reconfiguration of i/o resources can be done dynamically and in an automatic manner (Tarui's paragraph 84 lines 12-16);

Day does not expressly disclose the claim's aspect of ownership of I/O tasks. However, Tarui further discloses the plurality of logical partitions comprising at least one logical partition that owns identified I/O and at least one logical partition that does not own the identified I/O

(Tarui's paragraph 53, space sharing allocation allocates each partition with separate I/O resources; It's further noted that each logical partition can own and process its own identified I/O tasks in separate manner, thus a second logical partition would not process the I/O task that has been assigned, owned and identified by a first logical partition);

Day and Tarui do not expressly disclose the claim's details associating with I/O resources being arranged in I/O loops. However, Kanano discloses a storage subsystem with I/O resources that are arranged including a plurality of I/O towers (Kanano's Fig 2: #11 to #14, paragraphs 27-28) coupled to the apparatus via plurality of I/O loops (Kanano's Fig 2: #211 to #214, and #221 to #224 etc..arrays of disks are arranged in several I/O loops associating with towers Fig 2: #11 to #14, see Kanano's paragraphs 27-28 ). Kaneko further discloses an I/O reconfiguration mechanism including : detecting when at least one I/O loop is unbalanced (Kaneko's Fig 2, paragraph 38, when new disk resources being added into the storage subsystem, the new disk resources can be distributed to different users/different logical partitions, see Kaneko's paragraph 12 ); suspending all of the plurality of logical partitions by inhibiting dispatch of tasks to all of the plurality of logical partitions and waiting until all pending tasks in all of the plurality of logical partitions in all of the plurality of logical partition are completed ( In order to provide disk spaces of new disk resources to different users/different logical partitions, all the tasks in all of these different partitions must be further inhibited by inhibiting dispatch of tasks to all of these different partitions, and waiting until the pending tasks are completed; see Kaneko's Fig 4, paragraphs 27-28,35, the requests which are received from the host(s) by any other disk array unit that accessing the I/O resources of disk array units having added I/O resources are determined, disconnected and suspended, for example to add a disk in the disk

array unit 3; It's noted that the requests are readily coming from different users/different partitions, see Kaneko's paragraph 12); reconfiguring the I/O loop so the I/O loop is balanced (Kaneko's paragraph 38 further discloses an embodiment of rebalancing the I/O loop(s) with new I/O resources being added, by producing a plurality of copies of disk data and dividing them so as to utilize the disk regions connected to the newly installed disk array unit); Obviously after the configuring/rebalancing step, the new disks being added are ready for usage by enable the dispatch of tasks to all the different users/different partitions and resuming all the different partitions operations.

### *Response to Arguments*

Applicant's arguments in response to the last office action has been fully considered but they are not persuasive. Examiner respectfully traverses Applicant's arguments for the following reasons:

Regarding Applicant's remarks on pages 10-12 for the rejections of claims 1,4,9-11 and 19 under 35 U.S.C 103(a) as follows,

A) Regarding Applicant's remarks at pages 10-12 for the rejection of claim 4, Applicant arguments are clearly not understood, Applicant admitted that Kaneko teaches the suspension of all the logical partitions and somehow Kaneko's teaching of reconfiguration of I/O resources, (i.e disk arrays in Kaneko's Fig2) due to for example, adding one or more disks to provide more storage capacities for the whole system, would not read on the claim's limitations because the claim recites "at least one logical partition **that owns identified I/O** and at least one logical partition that **does not own the identified I/O**". Applicant then goes on to assert

“Kaneko teaches that all if the logical partitions are suspended but all of the logical partitions **own the resources** spoken of”, “..even if all of the partitions are suspended, only the partitions that own the resources have been suspended..” and “..stopping all the partitions in Kaneko only reads on stopping logical partitions that **own the identified I/O**”.

Firstly, There is nothing in the claim language that requires “a logical partition that own identified I/O resource, and another logical partition that **does not own identified I/O resource**” the claim’s language recite a logical partition that own the identified I/O and a logical partition that does not own the identified I/O”. If Applicant’s intention is directing specifically to two partitions, comprising of a logical partition does own the identified I/O resource and another partition does not own the same identified I/O resource”. Applicant must clearly state in the claim language. Instead the claim currently recites “owns identified I/O” and “not own identified I/O” which directs to owning the identified I/O operation

Secondly, Kaneko’s paragraph 32 discloses the detail logic of the I/O subsystem including memory storing information to maintain and routing all requests from all hosts to all disks properly, since disks of disk array unit can be partitioned to shared or not to be shared among different users/different partitions.

In the situation adding disks of the disk arrays to provide more storage capacities for the whole system, and then assigning capacity of new disks of the disk to not only current users but also to new different users/different partitions (Kaneko’s paragraph 38), it requires I/O operations in all partition be quiescent, the disks are added, reconfiguration and resuming the system to take advantage of newly configured expansion I/O storage capacity.

B) Regarding Applicant's remarks at page 12 for the rejection of claim 5,9-10,16,19. The claims are rejected based on discussion in above paragraphs.

*Conclusion*

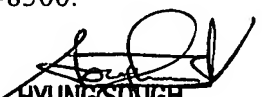
**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 36 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

When responding to the office action, Applicant is advised to provide the examiner with the line numbers and page numbers in the application and/or references cited to assist examiner to locate the appropriate paragraphs.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Duc T. Doan whose telephone number is 571-272-4171. The examiner can normally be reached on M-F 8:00 AM 05:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung S. Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

  
HYUNG SOUGH  
SUPERVISORY PATENT EXAMINER

7/11/07